



Utilizando INNER JOIN #1

Transcripción

[00:00] Hola. En las clases anteriores desarrollamos la última funcionalidad del reporte en nuestra aplicación y aprendimos cómo relacionar tablas con la creación de categorías para los productos. En nuestra entidad de productos nosotros agregamos una clave foránea y hace referencia a una de las categorías que tenemos registrada en la entidad categoría.

[00:20] Para crear el reporte, nosotros utilizamos una solución que no es muy buena, ya que genera n queries en la base de datos, porque nosotros aquí en el `cargaReporte` para cada categoría estamos yendo a la base de datos para listar a los productos de esta categoría, eso genera muchas conexiones con la base de datos.

[00:41] Y con el aumento del volumen de información y de utilización de la base de datos en la aplicación, eso puede generar problemas de performance más adelante. Vamos a la solución de este caso. Para cargar el reporte, nosotros creamos el método `cargaReporte` en la clase `reporteFrame` y en ella nosotros estamos buscando las categorías acá con el contenido para hacer un loop.

[01:06] Y por cada iteración de este loop nosotros estamos buscando los productos. La idea ahora de lo que podemos hacer es buscar a todas las categorías y productos en una sola query. Entonces vamos aquí a `cargaReporte` en la clase `categoríaController`, y en lugar de estar llamando el método `listar` que ya tenemos aquí que lista todas las categorías, nosotros vamos a crear otro método.

[01:32] Vamos a desarrollar un nuevo método del listado de estas categorías para el reporte. Entonces vamos a hacer lo siguiente. En lugar de estar devolviendo el resultado de listar, vamos a hacer así, vamos a hacer un return `this.categoríaDAO`. Ahora es la categoría dado que nos va a devolver esta información y vamos a crear un método llamado `listarConProductos`.

[01:58] O sea, este método lo que va a hacer es justamente lo que dice su nombre, es listar a las categorías con sus productos. La unión de estas dos tablas. Ahora vamos a crear este método acá en el `categoríaDAO`, ahí lo creamos y vamos a tomar como base lo que ya tenemos aquí en la lógica del método de listar.

[02:19] Entonces yo voy a copiar esta lógica acá ahora y ahí vamos a hacer los cambios de la query para poder dejarla de la forma que necesitamos, haciendo un join con la tabla de productos, entonces, ya copié el método. Vengo acá, lo pego, ahí está, voy a formatear. Listo, muy bien. El objetivo aquí ahora es cargar las informaciones de productos y las categorías punto en la misma query.

[02:46] En SQL, es posible hacerlo con un recurso de JOIN entre las tablas, más específicamente, el INNER JOIN. Ese recurso nos posibilita unificar dos tablas que tienen columnas en común y para nuestro caso la tabla producto tiene la columna categoría id, que hace referencia a la columna de id, a la de la tabla categoría.

[03:10] Entonces nuestra query aquí puede ser la siguiente, ya tenemos la base que es "SELECT ID, NOMBRE FROM CATEGORÍA"; lo que vamos a hacer ahora es darle un alias, un apodo a la tabla categoría. ¿Por qué? Porque para hacer el JOIN nosotros tenemos que identificar las tablas para poder saber de qué tabla es la columna que estamos haciendo la referencia.

[03:33] Entonces ahora estoy haciendo un CATEGORÍA C, entonces hacemos un SELECT C.ID y C.NOMBRE de la CATEGORÍA C, acá hacemos un "INNER JOIN

PRODUCTO P". Y tenemos que poner una condición, entonces la condición va a ser ON. C.ID = P.CATEGORIA_ID.

[04:00] O sea, estoy haciendo una referencia de la clave primaria de categoría que es su id, con la clave foránea que tenemos en PRODUCTO P, que es la CATEGORIA_ID. Y así hacemos el JOIN, ahí está, esta es nuestra query. Ya completamos, tenemos el JOIN entre las dos columnas. ¿Qué nos falta ahora? Devolver la información de productos.

[04:21] Entonces aquí en el SELECT hacemos C.ID, C.NOMBRE y ahora vamos a agregar P.ID, que es el ID del producto, P.NOMBRE y P.CANTIDAD que son las informaciones que nosotros estamos imprimiendo en nuestro reporte. Okay, voy a formatear un poquito más acá, ahora sí queda más linda la query. Voy a guardar todo y levantar la aplicación una vez más.

[04:48] O sea, voy a cerrar la aplicación que está acá levantada y la levanto una vez más para que probemos el resultado. Ahora vamos acá en nuestra aplicación, hacer un clic en ver reporte y mira qué pasó, tenemos muebles acá una vez, tecnología una, dos, tres, un montón de veces parece que ya rompimos acá la aplicación.

[05:12] Vamos a ver qué pasa, por qué no debería estar devolviendo tantos resultados así. O sea, voy a bajar acá la aplicación y lo que vamos a hacer es lo siguiente, vamos a copiar esta query en la consola de MySQL para entender qué está haciendo esta query y por qué devolvió tantos resultados. Entonces voy a copiar acá este contenido.

[05:31] Voy a llevar a la consola, acá está el SELECT, el FROM también. Y por último, el INNER JOIN. Lo estamos agregando también, ahora punto y coma y enter. Y mira qué está pasando acá. Estamos devolviendo todo el JOIN de categorías y productos. Tenemos acá muebles con la mesa, tenemos tecnología con el celular, la linterna, teclado y mouse. Cocina para vaso, cuchara, botellas y platos.

[06:02] Y las zapatillas. Y mira qué interesante eso. Nosotros estamos repitiendo la información de tecnología, entonces estamos mostrando tecnología una, dos, tres, cuatro, cinco veces, o sea, nuestro resultado de la query devuelve 12 filas para el reporte. Y las 12 filas acá en el reporteFrame, ¿qué estamos haciendo?

[06:29] Categoría. Porque estamos tomando las categorías todavía, entonces las 12 categorías estamos buscando otra vez a los productos en la base. Entonces aún tenemos que hacer algunas mejoras en este código para poder hacer que mostremos solamente la categoría una sola vez y los productos, la cantidad de veces necesaria, que es una por cada producto que tenemos relacionado con la categoría.

[06:53] Acá en la categoríaDAO de un método que creamos, vamos a arreglar la situación de la siguiente forma. Acá en el resultado, nosotros estamos asignando a las 12 filas con las categorías repetidas que hicimos con la query de INNER JOIN. Lo que vamos a hacer para resolver esta situación, para no repetir a la categoría, es la siguiente.

[07:14] Vamos a estar agregando las categorías en el listado y todas las veces que pasemos acá en el WHILE por cada categoría, vamos a ver si ella ya está agregada en el resultado. Si está agregada, nosotros la utilizamos y seguimos con el flujo. Si no está agregada, nosotros la creamos y la agregamos ahí en el listado.

[07:35] Entonces la lógica va a quedar más o menos así. Lo primero que vamos a hacer acá es extraer estas informaciones de resultSet en variables, entonces voy a tener acá la categoríaId. Y la próxima va a ser la categoríaNombre. Entonces hice la extracción, categoríaNombre y ya las tenemos acá como variables adentro de WHILE. Ya vamos a entender por qué estoy haciendo eso.

[08:08] La segunda parte que vamos a hacer en esta lógica es la siguiente. Esta categoría que estamos tomando acá, que estamos tomando la categoría, vamos

a hacer así, vamos a decir que esta categoría va a recibir un siguiente resultado, voy a dar un enter acá y vamos a hacer así, vamos a buscar en el listado de resultados.

[08:28] Vamos a hacer un resultado.stream, vamos a filtrar, hacer un filter de la categoría que está acá en la stream, en donde la cat.getId, o sea, el id de la categoría sea igual a categoríaId. Y si encontramos cualquier resultado de esa categoría con id igual, nosotros la vamos a utilizar.

[08:56] Si nosotros no lo encontramos, o sea, vamos a tener acá una condición de .orElseGet, si nosotros no encontramos la categoría, vamos a tener una función acá que vamos a hacer lo siguiente. Vamos a hacer un new categoría y vamos a agregarla al resultado. Entonces esta lógica que dejé abajo la pongo acá dentro y está new categoría, yo la voy a asignar a una variable local que va a llamarse cat, así como hicimos con el lambda de arriba.

[09:35] Y este cat, nosotros lo agregamos al resultado. Y al final de eso nosotros hacemos un return cat; ¿por qué? Porque lo vamos a asignar a esta variable de acá. Para entender un poco más lo que hicimos acá, primero porque hicimos la extracción de las variables para no utilizar adentro de stream.

[09:59] ¿Por qué? Porque en stream, cada lambda que tenemos acá es otro contexto y si nosotros estuviéramos utilizando el resultSet acá adentro de este contexto, nosotros íbamos a tener que agregar cada pedazo de código de acá adentro de las lambdas adentro de un bloque de try catch.

[10:17] Por eso que hice esta extracción, así nosotros ya aprovechamos el propio try with resources, que ya tenemos acá afuera del WHILE, que ya tiene todo eso adentro de su contexto. La otra parte de la lógica que estuve hablando fue la siguiente. Acá estamos tomando el resultado, que es un list de categoría.

[10:40] Lo estoy transformando en un string y haciendo un filter, o sea, estoy buscando si en este listado ya tenemos una categoría con este id de esta variable, de este loop que estamos haciendo. Si ya existe, o sea, un findAny si

encontramos cualquier resultado que tenga esta igualdad de la condición acá del filter, nosotros vamos a asignar este resultado a la variable de categoría.

[11:10] Si no existe nada de eso, o sea, es la primera vez que estamos pasando por esta categoría del id específico, entonces nosotros estamos creando el objeto de la categoría y lo estamos agregando a nuestro listado de resultados, vamos a ver cómo funciona ahora. Acá levanté la aplicación. Vamos a ver el reporte una vez más. Hice un clic acá.

[11:40] Y ahora sí estamos mostrando otra vez las categorías correctamente, o sea una sola vez, en lugar de estar repitiendo por cada fila que tenemos en el JOIN, pero seguimos acá ejecutando las queries $N + 1$. Aún tenemos que arreglar esta situación ya utilizando acá la información de producto que tomamos desde la query de JOIN.